The Balance of Secure Development and Secure Operations in the Software Security Equation

GFIRST 2010

Aug 17, 2010

Sean Barnum

Software Assurance Principal
sbarnum@mitre.org

UNCLASSIFIED

The Software Security Equation

- **Software security is about reducing the risk that software poses to those who use it or are affected by it.**

SS = Risk reduction

- **This requires thought and action more than simply at the point of development or use.**

SS != SD or SO

- **It requires a more holistic approach, balancing secure development and secure operations.**

SS = SD and SO

- **Bad news: these two capable domains typically do not interact much or understand each other.**

- **Good news: there are active ongoing efforts focused on addressing this gap.**

UNCLASSIFIED

Secure Development

- The objective of security in development is to prevent security issues in the software causing vulnerability.
- Best case, this means preventing such security issues from ever entering the software to begin with.
  - This best-case approach is driven by activities such as:
    - effective security training, security policy definition, security requirements specification and review, secure architecture and design, and architectural risk analysis.
- Worst case, this means at least preventing such security issues from ever being fielded into live systems.
  - This later life-cycle approach is typically driven by activities such as:
    - secure code analysis, security testing, and penetration testing.

Secure Operations

- The objective of security in operations is to prevent security issues in deployed systems by securing their *infrastructure, configuration, and use.*
- Ultimate goal would be to have all operating software totally free from vulnerability and fully secure.
- Given the complexities involved in today's software and the everchanging threat landscape, the reality is that no software can ever be presumed as *fully secure and will typically be under ongoing* and consistent attack.
- Beyond the initial security engineering of software operational deployment, the bulk of secure software operations is about continuous situational awareness and incident response.
- Recognizing real-world practicalities, it is focused on answering a set of foundational, ongoing secure operations questions

Foundational Questions of Secure Operations

- **Are we being attacked? (Were we attacked?)**
- **How are we being attacked?**
- **What is the objective of the attack?**
- **What is our exposure?**
- **Who is attacking us?**
- **What should we do to protect against these attacks in the future?**

Mechanisms of Secure Development

- Effective Security Training
- Security Policy
- Security Requirements
- Secure Architecture & Design
- Secure Coding
- Security Testing
- Penetration Testing

Mechanisms of Secure Operations

- Secure Configurations
- Firewalls
- Proxies
- Anti-Tamper (AT) Mechanisms
- Intrusion Detection Systems (IDS)
- Intrusion Prevention Systems (IPS)
- Real-time Data Monitoring
- Operational Monitoring and Control
- Incident Response
- Forensics

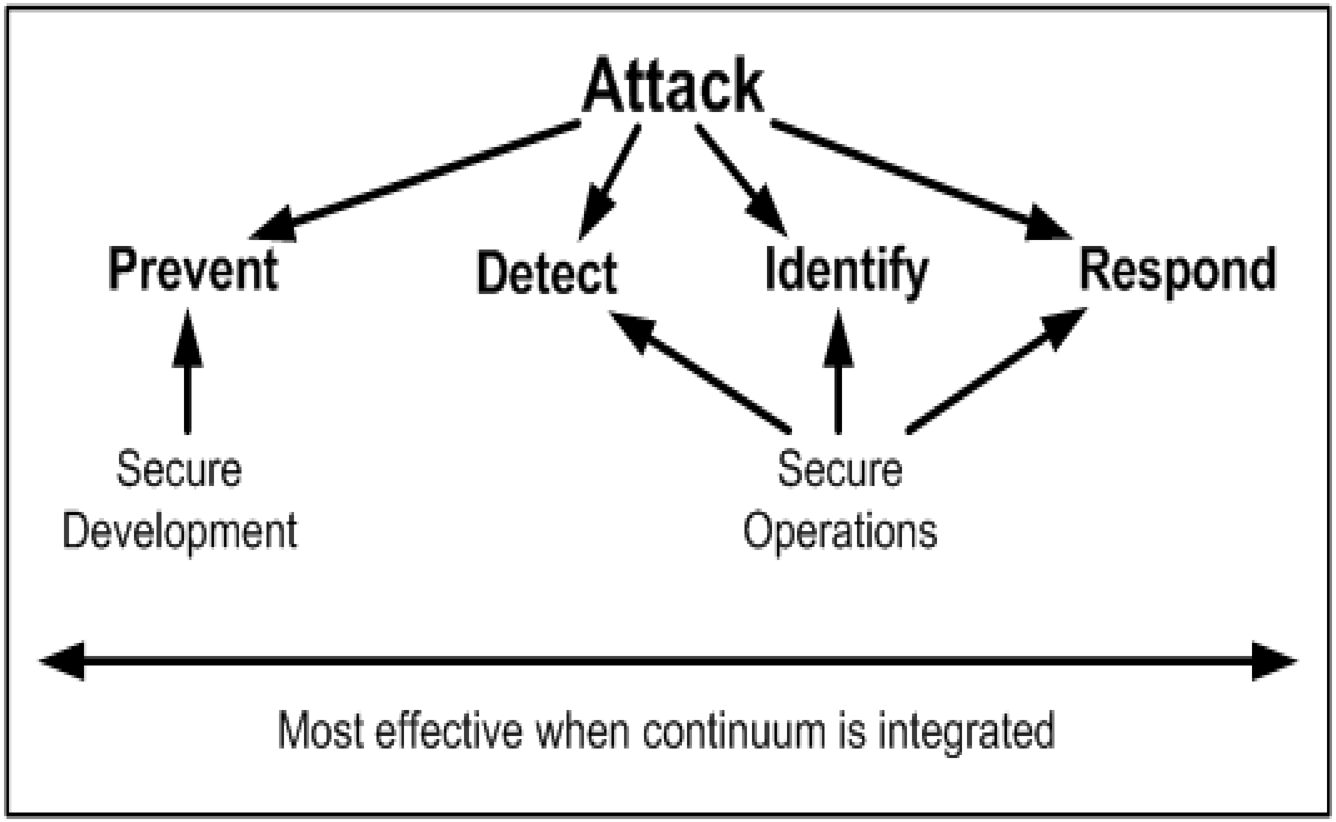
Commonality of Attack

- **The commonality between the secure development and secure operations domains is the central role of understanding how adversaries attack software.**

- **The secure development domain needs to understand the attacker's perspective in *abstract terms* in order to improve security across a wide range of contexts, rather than individual instances.**

- **The secure operations domain needs to understand the attacker's specific variations of behavior in *gory detail* in order to recognize it, understand it, estimate its effect, and plan its mitigation.**

- **Reciprocal balance between the top-down perspective of secure development and the bottom-up perspective of secure operations yields opportunity for mutual benefit.**


Attack
Prevent
Detect
Identify
Respond
Secure Development
Secure Operations
Most effective when continuum is integrated

Attack Patterns

- **Given the differing requirements between the two domains (to characterize attacks and potentially exchange this information), a flexible mechanism is required to capture, describe, and share knowledge about common patterns of attack.**
- **The attack pattern concept represents a description of common attack approaches abstracted from a set of known real-world exploits.**
- **Attack pattern object as specified and leveraged by the Common Attack Pattern Enumeration and Classification (CAPEC) - http://capec.mitre.org**

Common Attack Pattern Enumeration and Classification (CAPEC)

- **Community effort targeted at:**
  - **Standardizing the capture and description of attack patterns**
  - **Collecting known attack patterns into an integrated enumeration that can be consistently and effectively leveraged by the community**
  - **Gives you an attacker's perspective you may not have on your own**
  - **Initially, attack-centric testing methods, now integrating with operations and malware**
- **Excellent resource for many key activities**
  - **Abuse Case development**
  - **Architecture attack resistance analysis**
  - **Risk-based security/Red team penetration testing**
  - **Whitebox and Blackbox testing correlation**
  - **Operational network observation correlation**
- **Where is CAPEC today?**
  - **http://capec.mitre.org**
  - **Currently 311 patterns, stubs, named attacks**
  - **Future plans**
    - **New patterns**
    - **Align patterns with other resources**
    - **Formalize patterns to finer granularity to support bridging with the malware and incident response communities**

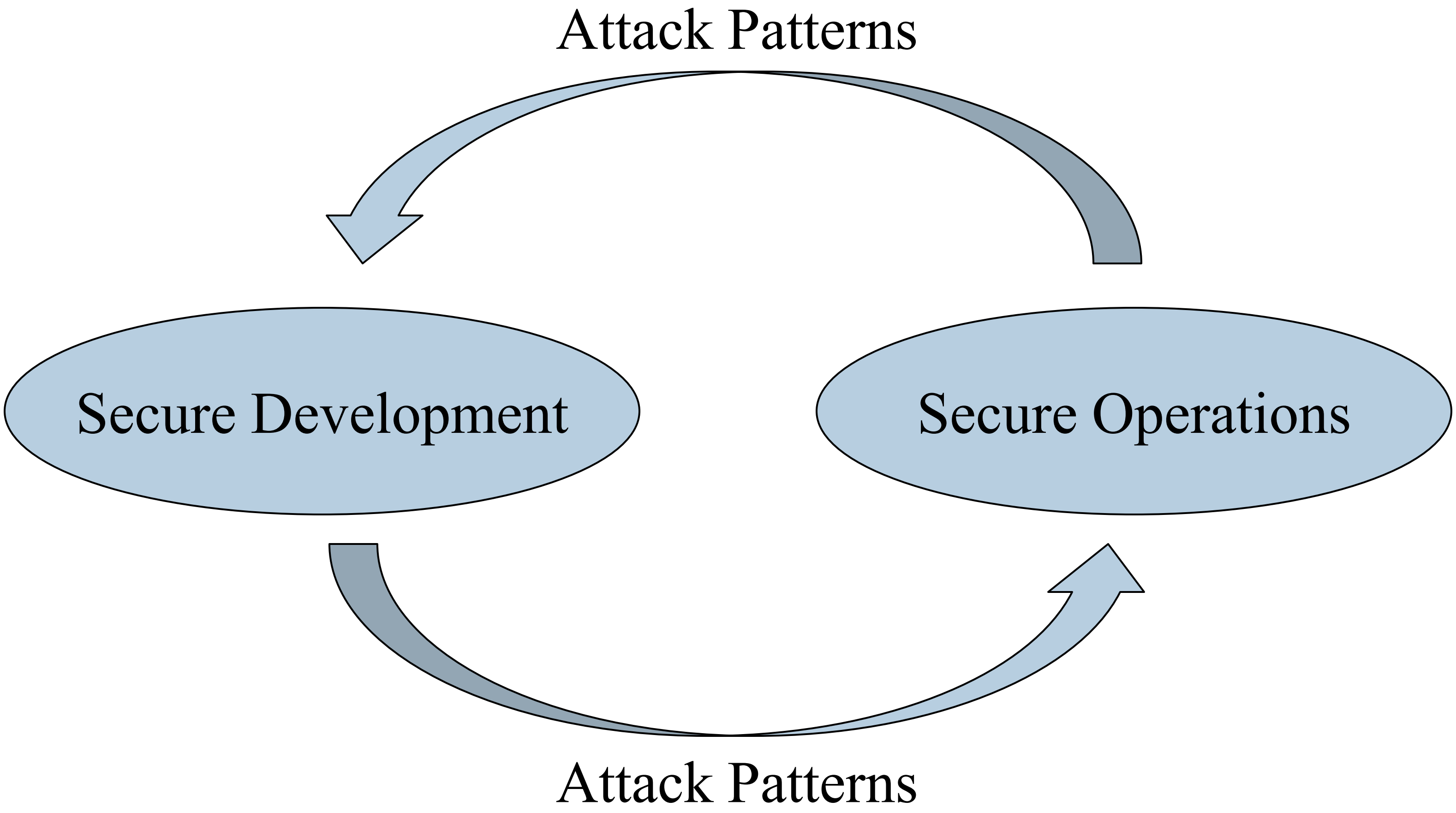
Attack Pattern Value to Secure Development

- **While this source of raw data comes primarily from the secure operations domain, attack patterns today are primarily a construct used by the secure development community to aid software developers in improving the assurance profile of their software.**
- **Attack patterns offer the secure development community unique value in several areas such as:**
  - **Representing abuse cases (how an attacker would intentionally abuse a software system) during requirements elicitation, specification, and review.**
  - **Mapping identified threats to the software's modeled attack surface as part of threat modeling activities during architecture and design.**
  - **Guiding and prioritizing secure code analysis during implementation. This includes identifying specific high-risk areas requiring greater analysis rigor as well as the most relevant weaknesses to look for.**
  - **Identifying, specifying, and prioritizing security test cases.**
  - **Serving as attack templates for penetration testing and objective persona descriptors for red team penetration testing.**

Attack Pattern Value to Secure Operations

- The secure operations community can utilize CAPEC to assist in situational awareness of deployed systems under attack and aid in response and mitigation.
- Several characteristics of attack patterns make them relevant for the secure operations community:
  - Attack patterns provide high-level rather than simply low-level detailed patterns of attacks against software. Much of secure operations is about analyzing low-level activity for patterns and composing them into higher levels of abstraction to detect, identify, and respond to attacks.
  - Software assurance attack patterns provide a top-down, high-level context for both the method and the intent of attacks.
  - Efforts are currently under way to formalize the CAPEC attack pattern schema in order to provide adequate detail of attacks for aligning and integrating their context with bottom-up incident analysis characterizations.

Attack Patterns Bridge Secure Development and Operations



Attack Patterns

Secure Development

Secure Operations

Attack Patterns

Secure Operations Knowledge Offers Unique Value to Secure Development

- Using attack patterns makes it possible for the secure development domain to leverage significant value from secure operations knowledge, enabling them to:
  - Understand the real-world frequency and success of various types of attacks.
  - Identify and prioritize relevant attack patterns.
  - Identify and prioritize the most critical weaknesses to avoid.
  - Identify new patterns and variations of attack.

Secure Development Knowledge Offers Unique Value to Secure Operations

- **It is also possible for the secure operations domain to leverage significant value from secure development knowledge as captured in CAPEC attack patterns.**

- **This enables those in the secure operations domain to provide appropriate context to the massive amounts of data analyzed to help answer the foundational secure operations questions.**

UNCLASSIFIED

Attack Patterns Help Answer Foundational Questions Regarding Secure Operations

Question	Role of Attack Patterns
Are we being attacked? (Were we attacked?)	Attack patterns offer structured descriptions of common attacker behaviors to help interpret observed operational data and determine its innocent or malicious intent.
How are we being attacked?	Attack patterns offer detailed structured descriptions of common attacker behavior to help interpret observed operational data and determine exactly what sort of attack is occurring.
What is the objective of the attack?	Elements of attack patterns outlining attacker motivation and potential attack effects can be leveraged to help map observed attack behaviors to potential attacker intent.
What is our exposure?	The structure detail and weakness mapping of attack patterns can provide guidance in where to look and what to look for when certain attack pattern behaviors are observed.
Who is attacking us?	Attack pattern threat characterization and detailed attack execution flow can provide a framework for organizing real-world attack data to assist in attribution.
What should we do to prevent against attacks in the future?	Attack patterns offer prescriptive guidance on solutions and mitigation approaches that can be effective in improving the resistance tolerance and/or resilience to instances of a given pattern of attack.

Maturing CAPEC to Better Support Automatable Integration of Both Domains

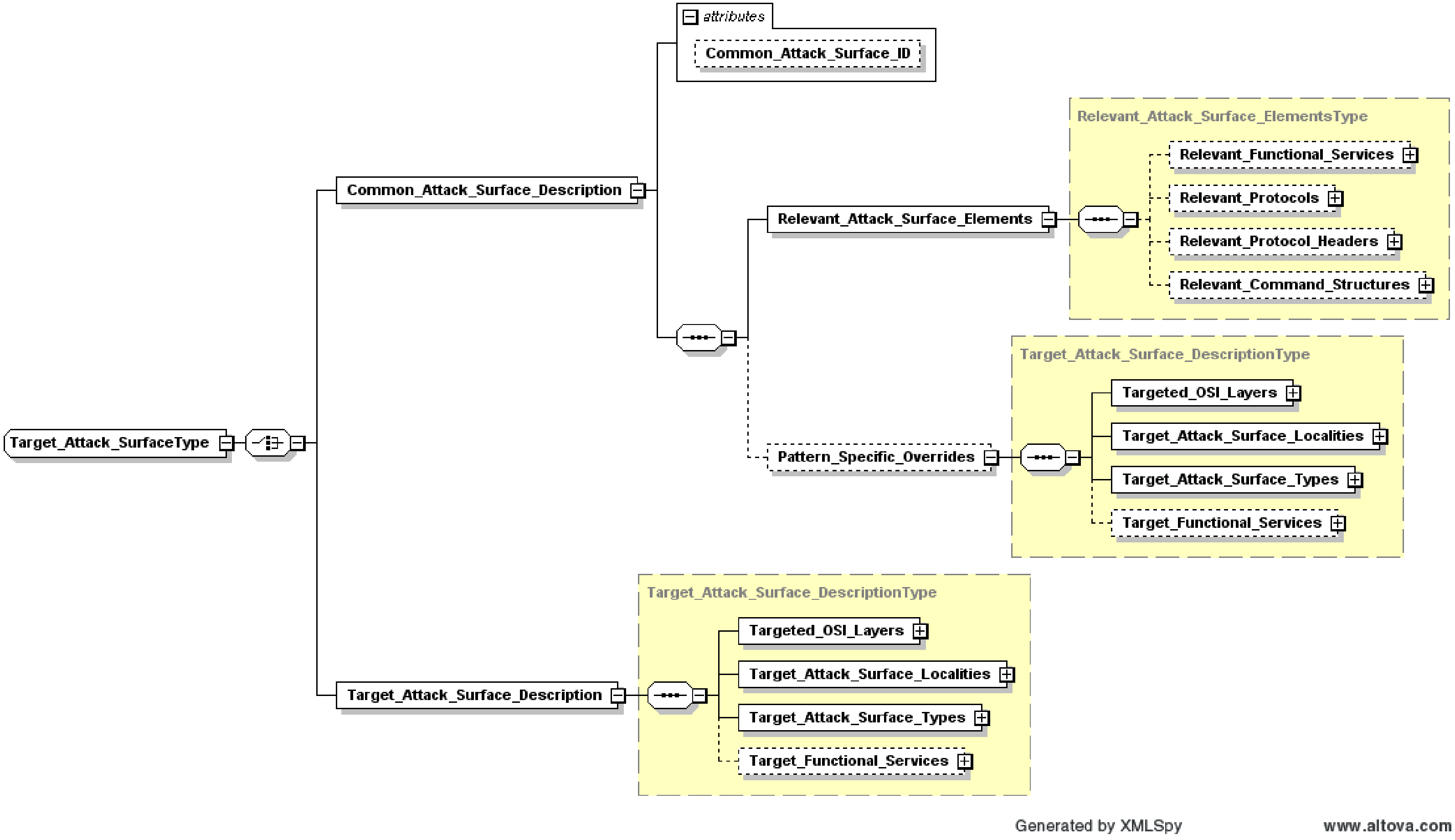
- **Currently focused on integrating and refining lower-level attack attributes and characteristics relevant to the secure operations domain.**

- **So far, this effort has been focused on enhancing attack pattern descriptions with greater levels of attack execution flow detail and on the addition of two new constructs: *Target_Attack_Surfaces* and *Observables*.**

Target_Attack_Surfaces

- **The Target_Attack_Surfaces construct is intended to give a structured characterization of the relevant portions of the targeted software that an attack is attempting to exploit.**
- **This sort of detail can be valuable within an operational context, assisting in *attack detection, identification, and characterization* through mapping of observed effects on target software assets and resources.**
- **The current initial draft schema is very limited and focuses on characterizing functional services, protocols, command structures, etc.**
- **Future schema revisions should extend this conceptual construct to address a broader set of attack surface characteristics.**

CAPEC Initial Draft Attack Surface Schema


attributes
Common_Attack_Surface_ID
Common_Attack_Surface_Description
Relevant_Attack_Surface_Elements
Relevant_Attack_Surface_ElementsType
Relevant_Functional_Services
Relevant_Protocols
Relevant_Protocol_Headers
Relevant_Command_Structures
Target_Attack_SurfaceType
Pattern_Specific_Overrides
Target_Attack_Surface_DescriptionType
Targeted_OSI_Layers
Target_Attack_Surface_Localities
Target_Attack_Surface_Types
Target_Functional_Services
Target_Attack_Surface_Description
Target_Attack_Surface_DescriptionType
Targeted_OSI_Layers
Target_Attack_Surface_Localities
Target_Attack_Surface_Types
Target_Functional_Services
Generated by XMLSpy
www.altova.com

UNCLASSIFIED

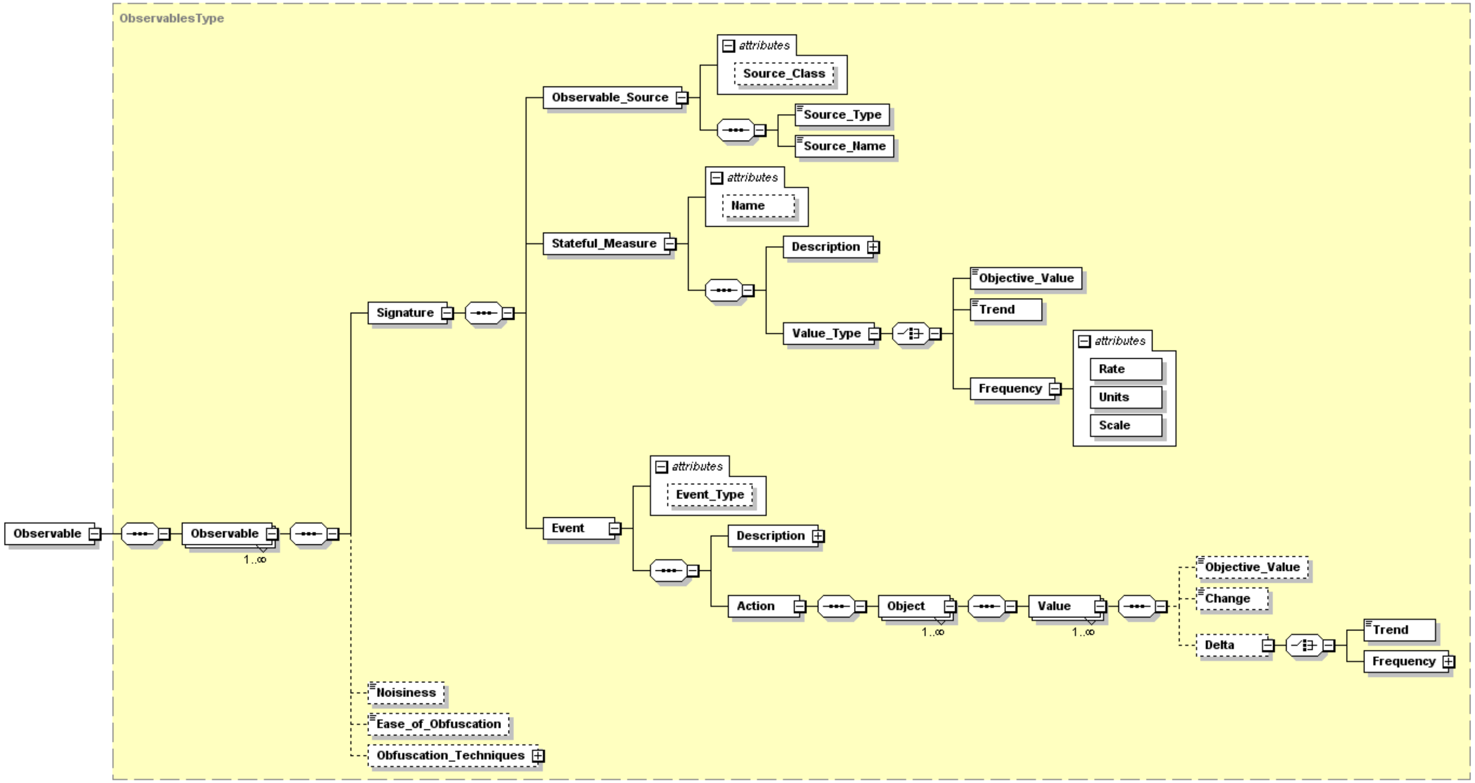
Observables Overview

- The Observables construct is intended to capture and characterize events or properties that are observable in the operational domain.
- These observable events or properties can be used to adorn the appropriate portions of the attack patterns in order to tie the logical pattern constructs to real-world evidence of their occurrence or presence.
- This construct has the potential for being the most important bridge between the two domains, as it enables the alignment of the low-level aggregate mapping of observables that occurs in the operations domain to the higher-level abstractions of attacker methodology, motivation, and capability that exist in the development domain.

Observables Intent

- By capturing them in a structured fashion, the intent is to enable future potential for detailed automatable mapping and analysis heuristics.
- The current Observables draft schema adorns the Attack_Step, Attack_Step_Technique, Attack_Step Outcome, and Attack_Step Security_Control elements of the attack pattern schema. It focuses on characterizing specific observable measures, their value, their sensor context, and how accurate or easy to obfuscate they are.
- Future schema revisions should flesh out the construct to cover other relevant dimensions.

CAPEC Initial Draft Observables Schema



Generated by XMLSpy www.altova.com

This schematic structure is currently undergoing significant revision and refinement. Changes will be based on input and collaboration from the operations community and other aligned knowledge standardization efforts needing this construct (e.g., Common Event Enumeration [CEE] and Malware Attribute Enumeration and Characterization [MAEC]).

A new version should be available in CAPEC v1.6 within the next month or so.

Summary

- Effective software security requires a balanced approach between secure development and secure operations.
- The commonality between these two domains is the central role of understanding how adversaries attack software.
- CAPEC attack patterns offer a mechanism for structured characterization of common attacks that enable a useful exchange of information relevant to both domains, also aligning low-level observations to high-level contexts for mutual benefit.
- CAPEC is currently a resource leveraged primarily by the secure development community, but there is an opportunity and a strong need for increased collaboration from the secure operations community.
- Collaboration from secure operations will help shape and refine CAPEC to more effectively serve both communities, potentially acting as an integrating bridge to eventually yield a more holistic software security capability.

Questions?

The topic and content covered in this presentation will be published as an article in the Sep/Oct 2010 issue of CrossTalk: The Journal of Defense Software Engineering